

Perancangan Mini Image Editor Versi 1.0 sebagai Aplikasi Penunjang Mata Kuliah Digital Image Processing

Erico Darmawan Handoyo

Program Studi Teknologi Informasi

Fakultas Teknologi Informasi, Universitas Kristen Maranatha

Jl. Prof. Drg. Suria Sumantri No. 65 Bandung 40164

Email: khe_wan_xing@yahoo.com

Abstract

Nowadays, there are a lot of graphic design softwares developed in multimedia technology. One which is very famous among them is Adobe Photoshop. Using this application, we can fix or process digital pictures to make them better. As one of the faculties in Computer Technology, Maranatha Faculty of Information Technology gives digital image processing subject to its students so that the students will know the basic processes behind digital image processing applications. A specific application have been built to provide the necessary basic skills for the digital image processing such as gray-scaling, bitmapping, brightness changing, blurring, and image scalling.

Keywords: Multimedia, Graphic design, Digital image processing, Basic proces, Adobe Photoshop.

1. Pendahuluan

Pada saat ini teknologi berkembang dengan pesat. Seiring dengan perkembangan teknologi tersebut, cara penyampaian suatu informasi/data yaitu berupa audio/visual dapat didigitalisasi. Salah satu penyampaian informasi/data yang berupa visual adalah melalui gambar.

Sebelum berkembangnya teknologi komputer, gambar hanya dapat direkam di dalam sebuah kertas film. Namun setelah perkembangan komputer yang cukup pesat, kini gambar-gambar tersebut dapat kita digitalisasi dan disimpan dalam media penyimpanan seperti harddisk, flashdisk, memory, dll.

Universitas Kristen Maranatha yang menyadari akan hal ini telah mengadakan mata kuliah Pengolahan Citra Digital (*Digital Image Processing*) pada kurikulum Jurusan Teknik Informatika. Mata kuliah ini bertujuan agar mahasiswa dapat memahami dan menerapkan dasar-dasar pengolahan citra digital. Dengan begitu, tidak hanya dapat memakai program-program pengolah citra seperti Adobe Photoshop tetapi juga dapat mengetahui proses-proses yang ada di baliknya.

Untuk dapat lebih memahami pelajaran ini, maka penulis mencoba membangun sebuah aplikasi yang bernama Mini Image Editor dan akan dipakai sebagai project pada mata kuliah tersebut. Dengan aplikasi ini, maka mahasiswa akan dapat memahami dan menerapkan ilmu-ilmu dasar pengolahan citra digital.

Beberapa fitur dari aplikasi ini yang akan dibahas adalah:

- a. Proses pengubahan gambar berwarna menjadi gambar grayscale.

- b. Proses pengubahan gambar berwarna menjadi gambar bitmap (hitam dan putih).
- c. Proses perubahan brightness suatu gambar.
- d. Proses Blur.
- e. Proses memperbesar suatu gambar.

2. Dasar Teori

Berikut adalah penjelasan untuk teori-teori yang diperlukan dalam mengembangkan aplikasi ini, yaitu:

2.1 Digitalisasi Citra

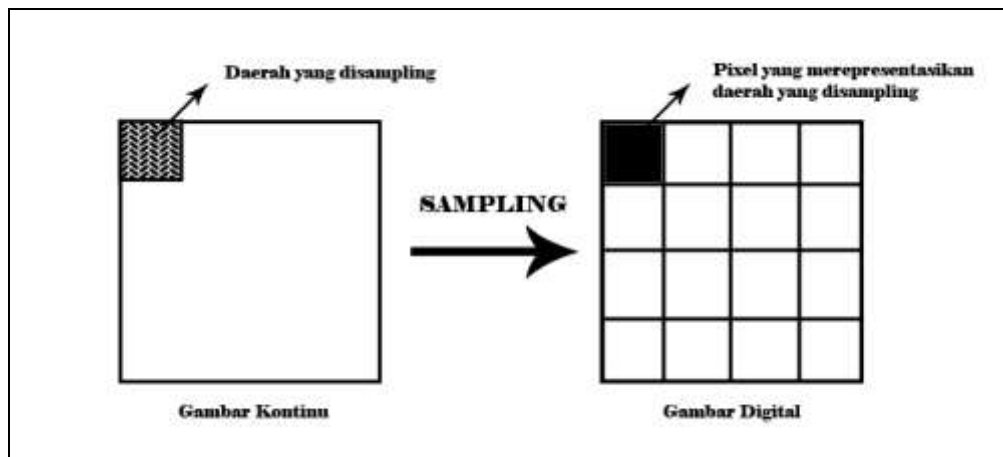
Agar dapat diolah dengan komputer digital, maka suatu gambar harus direpresentasikan secara numerik dengan nilai-nilai diskrit. Representasi gambar dari fungsi kontinu menjadi nilai-nilai diskrit disebut digitalisasi. Gambar yang dihasilkan inilah yang disebut dengan citra digital (digital image).

Proses digitalisasi gambar ada dua macam, yaitu:

- a. Digitalisasi spasial (sampling).
- b. Digitalisasi intensitas (kuantisasi).

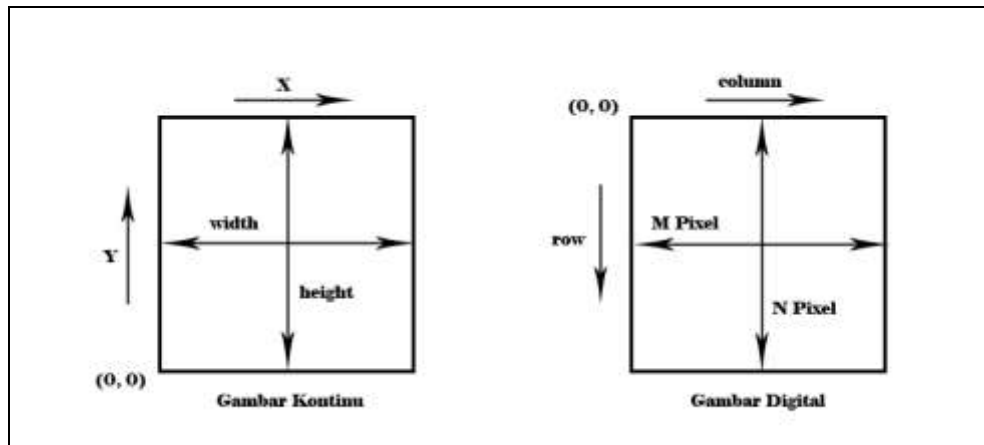
2.1.1 Sampling

Proses representasi suatu daerah dengan ukuran tertentu pada gambar kontinu ke dalam grid berbentuk bujursangkar disebut sebagai sampling. Grid tersebut adalah satuan terkecil dari gambar digital yang disebut pixel.



Gambar 1. Proses Sampling

Ada perbedaan koordinat gambar yang di-sampling dengan matriks hasil sampling. Koordinat (0,0) gambar yang di-sampling terletak pada sudut kiri bawah, sedangkan koordinat (0,0) matriks hasil sampling terletak pada sudut kiri atas.



Gambar 2. Perbandingan gambar kontinu dan matriks gambar digital

Jumlah pixel yang merepresentasikan tiap satu inci gambar kontinu disebut resolusi. Semakin tinggi resolusi (semakin banyak jumlah pixel untuk tiap inci gambar), semakin halus gambar yang diperoleh. Ini disebabkan jumlah informasi yang hilang karena sampling semakin sedikit.



Gambar 3. Gambar dengan 96 pixel/inch



Gambar 4. Gambar dengan 4 pixel/inch

2.1.2 Kuantisasi

Proses kuantisasi adalah proses membagi-bagi warna kontinu menjadi N buah warna. Biasanya N diambil dari perpangkatan 2.

$$N = 2^d$$

N = jumlah warna (derajat keabuan)

d = pixel depth / kapasitas yang diperlukan untuk menyimpan sebuah informasi warna (dalam bit).

N	Rentang nilai	Pixel Depth
2^1	0 ~ 1	1 bit
2^8	0 ~ 255	8 bit / 1 byte
2^{16}	0 ~ 65535	16 bit / 2 byte

Semakin besar pixel depth yang dipakai untuk merepresentasikan gambar kontinu, semakin bagus gambar digital yang akan dihasilkan karena jumlah warna yang dapat direpresentasikan semakin banyak.

Pada umumnya, gambar grayscale memiliki 256 level dan membutuhkan kapasitas 8 bit / 1 byte untuk tiap pixelnya.

Gambar biner hanya dikuantisasi pada dua level, yaitu 0 (hitam) dan 1 (putih). Tiap pixelnya hanya membutuhkan kapasitas 1 bit.



Gambar 5. Gambar biner (1 pixel depth)



Gambar 6. Gambar grayscale (8 pixel depth)

2.2 Elemen-Elemen Citra Digital

Gambar/citra digital memiliki sejumlah elemen-elemen dasar. Elemen-elemen inilah yang akan kita manipulasi dalam proses pengolahan citra. Elemen-elemen tersebut antara lain:

2.2.1 Color

Warna adalah elemen dasar gambar digital yang paling penting. Setiap gambar yang didigitalisasi kita representasikan dalam bentuk matriks warna (*matrix of color*).

Warna adalah persepsi yang dirasakan oleh system visual manusia terhadap panjang gelombang cahaya yang dipantulkan oleh sebuah objek. Tiap warna memiliki panjang gelombang yang berbeda. Warna merah memiliki panjang gelombang paling tinggi, sedangkan warna ungu memiliki panjang gelombang paling rendah.

Warna-warna yang diterima oleh mata manusia merupakan gabungan dari cahaya dengan panjang gelombang yang berbeda-beda. Kombinasi warna yang memberikan rentang warna paling besar adalah kombinasi warna merah (R), hijau (G), dan biru (B). Nilai R, G, dan B pada gambar RGB 8 bit adalah 0 sampai 255.

2.2.2 Brightness

Brightness adalah nama lain dari tingkat kecerahan/intensitas cahaya. Elemen ini menyatakan banyaknya cahaya yang diterima oleh mata. Elemen ini dapat dirasakan sebagai lampu penerang berwarna putih ketika kita melihat suatu benda. Semakin terang cahaya lampu tersebut (Tingkat kecerahan/brightness tinggi), benda yang kita lihat akan semakin putih. Semakin redup (Tingkat kecerahan/brightness rendah), benda yang kita lihat semakin gelap. Dan ketika tidak ada cahaya lampu (Tingkat kecerahan/brightness = 0), benda yang kita lihat berwarna hitam.

Elemen brightness ini dapat kita peroleh dengan menggunakan rumus:

$$I = 1/3 * (R + G + B)$$

I = nilai *brightness / intensity*

R = tingkat warna merah

G = tingkat warna hijau

B = tingkat warna biru

2.2.3 Saturation

Saturation adalah tingkat kepekatan/konsentrasi suatu warna. Semakin tinggi nilai saturasi, semakin pekat warna tersebut. Warna merah adalah warna dengan nilai saturasi tinggi. Warna merah muda adalah warna dengan nilai saturasi rendah. Warna putih adalah warna dengan tingkat saturasi 0.

Elemen saturasi dapat kita peroleh dengan menggunakan rumus:

$$S = 1 - ((3 / (R+G+B)) * \min(R,G,B))$$

2.2.4 Hue

Hue adalah nilai yang menyatakan corak/warna. *Hue* berasosiasi dengan panjang gelombang cahayanya.

Nilai hue dapat diperoleh dengan menggunakan rumus:

$$H = \cos^{-1} \frac{2R - G - B}{2\sqrt{(R - G)^2 + (R - B)(G - B)}}$$

2.3 Gambar Grayscale

Gambar *grayscale* adalah gambar yang hanya terdiri dari beberapa tingkat warna dari putih hingga hitam. Gambar *grayscale* 8 bit memiliki 256 tingkat warna abu-abu mulai dari putih hingga hitam.

Cara mengubah gambar RGB menjadi *grayscale* adalah dengan cara mengganti seluruh nilai RGB pixel-pixelnya menjadi rata-rata jumlah nilai RGB tiap pixel tersebut tersebut.

Jika kita memiliki sebuah pixel RGB dengan nilai RGB: 200, 40, 0, maka nilai RGB untuk pixel *grayscale*nya adalah: 80, 80, 80. Nilai 80 diperoleh dari $(200 + 40 + 0)/3$.

Jika *picture* adalah *matrix of Color* dengan ukuran 600x800 di mana *Color* memiliki field R, G, dan B, maka algoritma untuk mengubah gambar RGB menjadi *grayscale* adalah sebagai berikut:

```
Color picture[600][800];
int row, column, average;

for (row = 0; row < 600; row++)
    for (column = 0; column < 800; column++)
    {
        average = (picture[row][column].R + picture[row][column].G +
                    picture[row][column].B) / 3;
        picture[row][column].R = average;
        picture[row][column].G = average;
        picture[row][column].B = average;
    }
```

2.4 Gambar *Bitmap*

Gambar *Bitmap* adalah gambar yang hanya terdiri dari dua warna dasar yaitu hitam dan putih. Gambar *bitmap* dapat digunakan untuk mempermudah proses pengenalan pola (*Pattern Recognition*) pada suatu gambar.

Jika *picture* adalah *matrix of Color* dengan ukuran 600x800, maka algoritma untuk mengubah gambar RGB menjadi bitmap adalah sebagai berikut:

```
Color picture[600][800];
int row, column, average;

for (row = 0; row < 600; row++)
    for (column = 0; column < 800; column++)
    {
        average = (picture[row][column].R + picture[row][column].G +
                    picture[row][column].B) / 3;
        if (average < 128)
        {
            picture[row][column].R = 0;
            picture[row][column].G = 0;
            picture[row][column].B = 0;
        }
        else
        {
            picture[row][column].R = 255;
            picture[row][column].G = 255;
            picture[row][column].B = 255;
        }
    }
```

2.5 Merubah Nilai *Brightness* Suatu Gambar

Salah satu yang dapat kita lakukan untuk memperbaiki gambar digital yang kita miliki adalah mengubah nilai *brightness* dari gambar tersebut. Hal ini diperlukan jika gambar yang kita miliki tersebut terlalu gelap.

Berdasarkan rumus *brightness* yang dapat kita lihat pada bagian 2.2.2, kita dapat menyimpulkan bahwa peningkatan 1 point terhadap nilai *brightness* berarti menaikkan 1 point pada tiap nilai R, G, dan B gambar tersebut.

Jika *picture* adalah *matrix of Color* dengan ukuran 600x800 dan *point* adalah jumlah kenaikan pada nilai *brightness*, maka algoritma untuk mengubah *brightness* pada gambar tersebut adalah sebagai berikut:

```
Color picture[600][800];
int row, column, point;

for (row = 0; row < 600; row++)
    for (column = 0; column < 800; column++)
    {
        R = picture[row][column].R + point;
        G = picture[row][column].G + point;
        B = picture[row][column].B + point;

        if (R > 255)
            picture[row][column].R = 255;
        else if (R < 0)
            picture[row][column].R = 0;
        else
            picture[row][column].R = R;

        if (G > 255)
            picture[row][column].G = 255;
        else if (G < 0)
            picture[row][column].G = 0;
        else
            picture[row][column].G = G;

        if (B > 255)
            picture[row][column].B = 255;
        else if (B < 0)
            picture[row][column].B = 0;
        else
            picture[row][column].B = B;
    }
}
```

2.6 Blur/Image Smoothing

Terkadang gambar digital yang kita peroleh memiliki gangguan/*noise*. Gangguan tersebut biasanya didapat akibat sampling yang tidak baik atau akibat saluran transmisi (pada pengiriman data).

Blur/Image Smoothing bertujuan untuk mengurangi gangguan/*noise* tersebut. Salah satu metode *blur* adalah dengan menggantikan warna suatu pixel dengan rata-rata nilai pixel tetangganya.

Jika *sourcePicture* adalah *matrix of color* dengan ukuran 600x800 dan menyimpan gambar digital yang akan di-*blur*, sedangkan *targetPicture* adalah *matrix of color*

yang akan menyimpan hasil blur, maka algoritma untuk proses *blur* adalah sebagai berikut:

```
Color sourcePicture[600][800], targetPicture[600,800];
int row, col, row2, col2, totalPixel, R, G, B;

for (row = 0; row < 600; row++)
    for (col = 0; col < 800; col++)
    {
        R = G = B = 0;
        totalPixel = 0;
        for (row2 = row-1; row2 <= row+1; row2++)
            for (col2 = col-1; col2 <= col+1; col2++)
            {
                R = R + sourcePicture[row2][col2].R;
                G = G + sourcePicture[row2][col2].G;
                B = B + sourcePicture[row2][col2].B;
                totalPixel = totalPixel + 1;
            }
        targetPicture[row][col].R = R / totalPixel;
        targetPicture[row][col].G = G / totalPixel;
        targetPicture[row][col].B = B / totalPixel;
    }
}
```

2.7 Enlarge

Terkadang kita perlu untuk mengubah ukuran suatu gambar digital baik memperbesar atau memperkecil. Mengubah ukuran gambar akan mempengaruhi kualitas/ detil gambar. Memperbesar suatu gambar digital akan menyebabkan gambar tersebut terlihat kotak-kotak (ketajaman gambar berkurang). Memperkecil gambar tidak mempengaruhi ketajaman gambar tetapi akan tetap mengurangi informasi kedetilan gambar.

Jika *sourcePicture* adalah *matrix of color* dengan ukuran 300x400 dan menyimpan gambar digital yang akan diperbesar sebesar 2x, sedangkan *targetPicture* adalah *matrix of color* dengan ukuran 600x800 yang akan menyimpan hasil pembesaran gambar, maka algoritma untuk proses *enlarge* adalah sebagai berikut:

```
Color sourcePicture[300][400], targetPicture[600,800];
int row, col, row2, col2;;

for (row = 0; row < 300; row++)
    for (col = 0; col < 400; col++)
    {
        targetPicture[row*2][col*2]=sourcePicture[row][col];
        targetPicture[row*2][(col*2)+1]=sourcePicture[row][col];
        targetPicture[(row*2)+1][col*2]=sourcePicture[row][col];
        targetPicture[(row*2)+1][(col*2)+1]=sourcePicture[row][col];
    }
}
```

3. Kesimpulan dan Saran

Beberapa kesimpulan yang dapat diambil dari penelitian ini adalah:

1. Aplikasi ini dapat dijadikan project di dalam mata kuliah Pengolahan Citra Digital. Yang perlu dilakukan oleh mahasiswa adalah mengisi setiap function yang sudah dikosongkan terlebih dahulu.
2. Aplikasi ini dapat membantu mahasiswa untuk lebih mengerti proses-proses dasar Pengolahan Citra Digital.

Aplikasi ini dapat dikembangkan lebih lanjut dengan menambahkan proses-proses Pengolahan Citra Digital lainnya seperti *edge detection*, meratakan *histogram*, *Image Smoothing* dengan median, dll.

Daftar Pustaka

- | | |
|---------|---|
| [Mun04] | Munir, R (2004). Pengolahan Citra Dijital dengan Pendekatan Algoritmik. Penerbit Informatika Bandung |
| [Tat02] | Tatsumaki (2002). Mastering Computer Graphics untuk Pemula. Nexx Media, Inc. |
| [You98] | Young, IT. Gerbrandts, J.J. van Vliet, L.J (1998). Fundamentals of Image Processing. Delft University of Technology, Netherlands. |